Caliber API Documentation

(Version 2.0)

(Revised: Apr-16)

Contents

Introduction

Caliber API version 2.0 is a Web API 2, REST*ful*, JSON-based interface. This new version is a dramatic shift from the previous WCF-based services and will standardize the way in which all integrations shall be made with Caliber.

Partnering with Caliber

One of our company's core foundations is building a robust and innovative 3rd party software vendor community around Caliber. We believe that real innovation comes into an industry from vendors who specialize in a particular niche. While we like to think we are innovators too, it's unrealistic to think that Caliber can be everything to everyone. That's why we're dedicated to providing and supporting this API. We want to make it easy to use, available at no cost, and something that encourages new integration partners to join forces with Caliber, filling gaps or providing customers with better, alternative solutions.

If you ever have any questions or requests, don't hesitate to contact us. In fact, we enjoy working with integration partners and communicating with them frequently. It helps us to better understand your products and specifically your integrations, and it helps you to stay in touch with new features in Caliber and the Caliber API.

Confidentiality

Caliber Software deems the information within this documentation and the methodologies described herein to be highly confidential and would harm the company if this information was shared with its competitors or other outside parties. Having access to this document means you or your company has signed a Non-Disclosure Agreement (NDA) with Caliber Software. Every care should be taken to ensure this document, and the information contained herein is only made available to people covered under the NDA and only those who have a "need to know". If you have been provided access to this document and neither you nor your company have signed an NDA with Caliber Software, you are instructed to destroy this document and all copies (printed and electronic) immediately.

Prior Versions

Prior versions of the Caliber API are now deprecated and will be removed from service on **January 1, 2017**. Support and feature requests shall only be considered for version 2.0 going forward. All integration partners are encouraged to update their Caliber integrations to accommodate this new API methodology. This version represents a substantial shift in technology, so you would be wise to not wait too long to begin making the transition.

What's New

Prior versions of the Caliber API were built up as integration partners needed certain data from Caliber; a bottom up approach. Version 2.0 takes more of a top down approach where as much data as possible from Caliber can be accessed and/or updated. Not only does this approach enable integration partners to access more data and Caliber functionality, it is also more efficient in the way data can be requested and delivered.

HATEOAS

The Caliber API implements (or at least tries its best to implement) a methodology called HATEOAS (Hyperlinks As The Engine Of Application State). Information about HATEOAS can be found at the following link: https://en.wikipedia.org/wiki/HATEOAS. In a nutshell, the API responses may include links that relate the response data to other related data. For example:

```
/api/v2/mgmtco
```

will return information about the management company, which includes the following link:

```
{"Links":[{"rel":"Logo", "href":"/mgmtco/logo"}]}
```

From this link information, integration partners will know that:

```
/api/v2/mgmtco/logo
```

will return the logo for that management company.

In another example, an integration partner may make the following call:

```
/api/v2/client/64
```

This will return information about the association specified by ClientID = 64. Included in the response might be the following link:

```
{"Links":[{"rel":"Manager", "href":"/client/64/manager"}]}
```

This lets the integration partner know that the following call would return information about the manager of the specified association:

```
/api/v2/client/64/manager
```

HATEOAS enables integration partners to request some of the data (address, phone number, etc.) and then retrieve related data only when needed. This is a much more efficient (and faster) process since the API only needs to response with the minimum amount of data required.

Versioning

The new version of Caliber API also implements versioning within the API. This greatly simplifies the ability to control versions when adding new functionality, yet needing to support older integrations. This version is v2.0 so all calls will be made to:

```
…/api/v2/…
```

As newer versions of the API are released, integration partners using the newly added features might use the following call:

```
…/api/v2.1/…
```

Unfortunately, the Caliber API v2.0 does not support previous version, but going forward, newer versions will support previous ones (back to v2.0).

Caliber Facts & Terminology

This section is provided to help Caliber API users understand some facts about Caliber and the terminology used in Caliber and by Caliber customers.

All Caliber Users Are On the Same Version

Caliber Software has a policy of "no customization". While we take our customers' feature requests seriously and incorporate them into the product, these features are added to everyone's instances of Caliber. Because of this, we know that each of our customers (and their users) are on the exact same version of Caliber, provided they are current with their support and maintenance plan (at the time of this writing, all active Caliber customers are current on their support and maintenance plans).

This is an important fact for 3rd party integrators since it means that the code you build to integrate your product with Caliber, for one customer, will work for all Caliber customers.

Caliber Update Schedule

Caliber Software currently updates the Caliber Desktop software every other month. The day of the month in which an update occurs is always the Friday between the 20th and the 26th. Occasionally, we push out updates to resolve issues that simply can't wait until the next update.

Updates to other Caliber components, such as the Caliber API, are made on an as-needed basis and are usually, but not necessarily, coordinated with a Caliber Desktop update.

Caliber's update process is automatic. Within an hour of an update being published, all customers' servers and databases are updated, and as each user logs into Caliber, their desktop software gets updated.

Terminology

The following is a list of some common terminology used in Caliber:

"Customer" – a customer refers to a Caliber Software customer, which could be a property management company or a self-managed HOA.

"Client" – a client refers to an association, which may be one of many associations managed by a Caliber customer. In the case of a customer who is a self-managed HOA, the “customer” will have only one client and that client will be the same as the customer.

"Group" – a group refers to a collection of dwellings within a client. For example a large association may have many subdivisions, tracts, phases, etc., which are referred to in Caliber as groups. Each client must have at least one group.

"Division" – a division refers to a collection of clients. Larger management companies may wish to group their associations by location, by manager, by type, etc. Divisions are for organizational purposes only and are not required in Caliber, nor do they have any impact on accounting. It is possible to nest divisions within divisions.

"Unit" – a unit refers to an individual dwelling.

"Unit Account" – a unit account refers to an account, whether current or previous, that is associated with a unit. The unit account is where all owners, occupants, accounts, and other things like vehicles and passes are linked. Because Caliber uses the unit account in its data hierarchy, historical information is maintained. In other words, if a home is sold, the current unit account (and all the data associated with it) becomes a previous unit account, and then a new current unit account is created for the new owners. Each unit will have 1 current unit account and n-number of previous unit accounts. A unit account is sometimes referred to (verbally) as an "owner account".

"Contact" – a contact refers to a person associated with a unit account. It could be an owner, occupant, entity, or 3rd-party (e.g. realtor). There is no limit to the number of contacts that can be associated with a unit account. A contact record is actually comprised of Person1 (e.g. husband) or Person2 (e.g. spouse), so in fact, 2 people can be contained in a single contact record. A contact can also be an entity (e.g. builder, trust, etc.) instead of 1 or 2 people.

"Client Contact" – refers to a person associated with the client. It could be a board member, committee member, or member of some other type of grouping of people that the association wishes to maintain.

"Transaction Account" – refers to an owner ledger. Typically owners have a ledger for their association dues. Some customers wish to, and some states mandate that there be separate ledgers for things like fines, architectural fees, etc. These multiple ledgers are referred to as transaction accounts. In the case of multiple transaction accounts, Caliber customers can designate one account (typically the assessments account) as the default account.

Data Structure Considerations

The following is provided to assist Caliber integrators in understanding the data structure and hierarchical concepts built into Caliber.

- Divisions are not required.
- A client may belong to 0 or 1 division.
- A client may belong to another client, such as in the case of master and sub associations.
- At least one group is required per client.
- A client can have n-number of groups.
- A unit must belong to one and only one group.
- A unit must have at least one unit account.
- Only one unit account may be "current" for each unit. Non-current unit accounts will have the 'IsCurrent' flag set to false and have an End Date, whereas current unit accounts will have the 'IsCurrent' flag set to true and have only a Start Date (no End Date).

- Contacts are linked to unit accounts, not the units directly. The links between contacts and unit accounts also provide information about the type of contact, such as owner, primary owner, occupant, primary occupant, renter, or 3rd party.
- A contact may be associated with one or more unit accounts provided all unit accounts are within the same client. This capability is referred to as "multi-unit owners".
- Data in Caliber is not deleted, but rather the 'IsDeleted' flag, which is a column in most tables, is used to indicate whether or not a user has deleted the record.
- Most tables in Caliber have 'DateCreated', 'CreatedBy', 'LastModified', and 'ModifiedBy' columns in order to track who did what when. This is also useful to Caliber integrators to be able to process changes made within Caliber.

Caliber API Standards

The Caliber API incorporates the following standards, assumptions and practices.

Error Messages

The Caliber API uses HTTP Response Messages to deliver any error messages. It overrides the HTTP 500 – Internal Server Error or HTTP 404 Not Found Error with a more appropriate reason phrase. For example, if trying to retrieve the management company's logo and the logo does not exist in Caliber, the service would return:

HTTP 404 – Logo not found

For errors associated with information contained in the Authorization Header, the API overrides the HTTP 401 – Unauthorized error and provides a more appropriate reason phrase.

The Caliber API will respond in one of 5 ways:

- HTTP 200 – OK
- HTTP 400 – Bad Request
- HTTP 401 (Unauthorized) – [error message]
- HTTP 404 (Not Found) – [error message]
- HTTP 500 (Internal Server Error) – [error message]

Be sure to check for these responses where applicable.

Request and Response Objects

Unless otherwise specified, all request and response data will be in JSON format (see http://json.org/ for more information). The Content-Type of responses will be "application/json". If the response is a document or image (upload or download), the Content-Type will be “application/octet-stream”.

Security

The Caliber API offers several layers of security to protect Caliber data and prevent unauthorized access.

API User Setup in Caliber

Before a Caliber integrator can access the Caliber API, it must obtain the following information:

- API Endpoint – this is the base URL of the Caliber API for the specific instance of Caliber.
- APICode
- APIUsername
- APIPassword

The combination of APICode, APIUsername and APIPassword uniquely identifies the Caliber integrator to a specific instance of Caliber. Caliber logs most API activity and links the APICode to such activities.

APICode, APIUsername, and APIPassword for each Caliber integrator are configured within Caliber ('Admin', 'API User Configuration' menu option). Caliber customers are responsible for setting up this information in Caliber and providing integration partners with their APICode, APIUsername and APIPassword. Customers don't typically know the API Endpoint, however this information can also be obtained from the 'API User Configuration' screen or by contacting Caliber Support at support@calibersoftware.com or (480) 699-3621.

Caliber customers can also restrict integrators by allowing them access to only certain areas (interfaces) and to only certain clients.

NOTE: It is incumbent upon the Caliber customer to create your APICode, APIUsername, and APIPassword in Caliber. It is not the responsibility of Caliber personnel to create this for them. Documentation on how to properly set up and configure an API user in Caliber can be found here:

http://support.calibersoftware.com/Documents/Caliber%20API%20User%20Administration.pdf

Authorization Header

The APICode, APIUsername and APIPassword provided to you by the Caliber customer are crucial for interacting with the Caliber API. Every call to the API interface (except 'Time' and 'Ping') must include an authorization header with this information. The authorization header must be in the following format:

"basic [base64-converted security string]"

Security String

The "security string" is in the following format:

[APICode]:[APIUsername]:[APIPassword] (without the brackets)

For example, if your APICode = "myAPICode", your APIUsername = "myUsername", and your APIPassword = "myPassword", then your "security string" would be:

"myAPICode:myUsername:myPassword"

This should then be converted to Base64 encoding. For example, the above security string would be encoded to:

"bXlBUElDb2RlOm15VXNlcm5hbWU6bXlQYXNzd29yZA=="

Prepend the base64-encoded string with the word "basic" and a space, and then write this to the Authorization Header:

"basic bXlBUElDb2RlOm15VXNlcm5hbWU6bXlQYXNzd29yZA=="

As mentioned above, this authorization header is required for all calls made to the Caliber API, and must contain the APICode, APIUsername, and APIPassword information at a minimum.

Pass-Thru Authentication

In Caliber API version 1, there was a concept of "global authentication" and "pass-thru authentication”. In essence, "pass-thru" authentication allows integration partners to pass thru the username and password of the person using the integration that matches the username and password stored within Caliber, thus providing context of the owner or board member. For example, if you wanted homeowners to be able to log into your website product, but use Caliber's usernames and passwords instead of having to create and maintain them in your product, you could do so using "pass-thru" authentication.

Most integration partners used the "global" authentication methodology because they have either maintained their own login processes, or don't consume data that requires an owner or board member context.

In the new Caliber API, if an owner (or board member) context is required, the owner's (or board member's) Caliber username and password should be appended to the security string within the Authorization Header as follows:

[APICode]:[APIUsername]:[APIPassword]:**[OwnerUsername]:[OwnerPassword]** (without the brackets)

For example, if the owner's username is "JohnDoe123" and his password is "jdPass1", then the full security string would look like:

"myAPICode:myUsername:myPassword:JohnDoe123:jdPass1"

The encoded authorization for the above example would be:

"basic bXlBUElDb2RlOm15VXNlcm5hbWU6bXlQYXNzd29yZDpKb2huRG9lMTIzOmpkUGFzczENCg=="

The security string must contain the APICode, APIUsername and APIPassword. The user's username and password are optional and even if provided, are only used if there is the appropriate owner or board member context.

For example, when calling the "clientlist" interface, if the owner's username and password are not provided, then the interface will return all active clients. But if the username and password are provided, then only the client to which that owner or board member is associated will be returned.

SSL Encryption

With prior versions of the Caliber API, SSL encryption was optional. With version 2.0, SSL encryption is now mandatory. Please take this into consideration if updating your integrations from the previous API version.

If a request is made of the API that uses “http” instead of “https”, IIS will use the URL Rewrite module to redirect the request to “https”.

API Sandbox

Caliber Software maintains an instance of Caliber known as the API Sandbox. This is intended for 3rd party integrators who wish to develop and test their integrations within a safe, non-production environment. The API Sandbox is updated just like all production databases, so you can be assured that you're always working with the latest version.

Access to the API Sandbox is provided by Caliber Software once a mutual NDA is executed. The endpoint for the sandbox is:

https://asp.reefpt.com/capi2_APISandbox

APICode, APIUsername, and APIPassword to use in the sandbox will be issued by Caliber Software.

Caliber Desktop Software

Being a Caliber integration partner entitles you to download and install the Caliber Desktop software. Having access to this software enables you to test your integration to ensure data is pulling thru the API as anticipated. The Caliber Desktop software is to be used only for the purposes of testing, training and implementing your integrations with the Caliber API.

The Caliber Desktop software can be downloaded at the following link:

http://asp.reefpt.com/software/caliber_desktop.zip

Once installed, you will need to obtain connection information and login credentials from Caliber Software.

The Caliber Support Team is available to provide limited training to familiarize you with the Caliber Desktop software. Please contact support at support@calibersoftware.com or (480) 699-3621 if you would like to request training and they will schedule it on a space-available basis.

Documentation, Testing Tools, and Utilities

The following is information about how the Caliber API is documented and about important tools and utilities that may be useful in development and testing your integration with the Caliber API.

Swagger

The Caliber API implements a documentation tool called Swagger. It enables Caliber developers to embed documentation within the API coding itself and make that documentation available to our integration partners. There is nothing you need to install in order to take advantage of Swagger, you simply need to click on the following link:

https://asp.reefpt.com/capi2_APISandbox/swagger

As you can see, a list of Caliber API interfaces is listed in your browser and are grouped by category. You can then expand each category and see all the interfaces that are available, and then expand each interface to see documentation on how to use it. There is even a “Try it out!” button, however because of our security methodology, this button will return a 401 (Unauthorized) error for all but the 'Time' and 'Ping' methods.

Maintaining API documentation is very challenging, especially in an environment where lots of changes are being made. Swagger enable us to keep current on this documentation since it provides real time information about the API methods that have actually been implemented.

Postman (Chrome plug-in)

Postman is a fantastic tool for testing the Caliber API. It is a Chrome plug-in product that can be downloaded and installed for free at the following link:

https://www.getpostman.com/

What's so nice about Postman is the ability to create, save, categorize, and share your test cases. Caliber is creating a collection of these test cases that can be used on the API Sandbox that uses a generic API User account for authorization. You can modify the authorization to your specific API User Account credentials if you like, but this collection will instantly give you the ability to test connectivity, and get started on your integration project. This API Sandbox collection for Postman is a work in progress and will be expanded ultimately to include test cases for each API interface. This collection can be downloaded at:

https://www.getpostman.com/collections/cc55aae3577a29f4e670

Telerik Fiddler

Fiddler is a wonderful tool and can be used to test your integrations with the Caliber API. It is available for free at the following link:

https://www.telerik.com/download/fiddler

Caliber Software does not provide support or training for Postman, Fiddler, or any other 3rd party development and testing tools. However, there are plenty of online tutorials if you are not already familiar with these tools.

Postman and/or Fiddler are also very helpful to us if you encounter a problem and need our assistance. Emailing your Postman or Fiddler session would enable us to quickly reproduce the problem you're encountering.

Sample Applications and Utilities

Caliber Software may from time to time, create and publish sample applications and utilities that might help you understand how to implement certain aspects of the Caliber API. If interested, please contact Caliber for more information.

Interfaces

Introduction

The interfaces available via the Caliber API are documented via Swagger (see above, or click this link: https://asp.reefpt.com/capi2_APISandbox/swagger). For organizational purposes only, the interfaces are grouped into the following categories (listed alphabetically):

- Billing Records
- CC&Rs
- Client Contacts
- Client Contact Groups
- Clients
- Common Areas
- Contacts
- Family Members
- Groups
- Invoices
- Maintenance
- Management Company
- Miscellaneous
- Other Payees
- Payments
- Pets
- Unit Accounts
- Units
- User Credentials
- Vendors
- Violations

Interface Endpoints

The complete URL for each interface is made up of the following components:

[API Endpoint]/[Route]

Route includes:

- The word "api",
- The version, and
- The service

For example, a call to the Management Company interface on Caliber's API Sandbox would be:

https://asp.reefpt.com/capi2_APISandbox/api/v2/mgmtco

where:

- "https://asp.reefpt.com/capi2_APISandbox" is the endpoint, and
- “api/v2/mgmtco” is the route
    - the word “api” (required),
    - “v2” is the version, and
    - “/mgmtco” is the service.

Another example:

https://asp.reefpt.com/capi2_APISandbox/api/v2/client/64/manager

where:

- "https://asp.reefpt.com/capi2_APISandbox" is the endpoint, and
- “api/v2/client/64/manager” is the route, where:
    - the word “api” (required),
    - “v2” is the version, and
    - “/client/64/manager” is the service.

API Modification Requests

The Caliber API is very much a living, breathing interface that can be modified by Caliber Software to meet specific needs of integration partners. Similar to our “no customization” policy for Caliber, we don’t customize the Caliber API for any one integration partner. Instead, any feature requests or modifications will be available to all integration partners. If you would like to request new interfaces or modifications to existing ones, please contact Caliber Software and make your request known.